Requested Patent: WO0169843A2

Title:

METHOD AND SYSTEM FOR COORDINATING SECURE TRANSMISSION OF INFORMATION ;

Abstracted Patent: WO0169843 ;

Publication Date: 2001-09-20 ;

Inventor(s): MCNAMARA TONY ;

Applicant(s): ABSOLUTEFUTURE INC (US) ;

Application Number: WO2001US07767 20010312 ;

Priority Number(s): US20000188657P 20000310 ;

IPC Classification: H04L9/08 ;

Equivalents: AU5082401 ;

ABSTRACT:

A method and system for coordinating transmission between a sender and a recipient. In one embodiment, a third party coordinates the distribution of session keys (e.g., symmetric keys) for sender and recipient. A sender computer may generate a session key, encrypt it using a public key of the recipient, and send the encrypted session key to the third party. The third party then forwards the session key to the recipient computer. Upon receiving the session key, the recipient computer decrypts the session key so that the sender and recipient can communicate using a message encrypted with the session key.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 September 2001 (20.09.2001)

PCT

(10) International Publication Number
WO 01/69843 A2

(51) International Patent Classification[7]: H04L 9/08

(21) International Application Number: PCT/US01/07767

(22) International Filing Date: 12 March 2001 (12.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/188,657 10 March 2000 (10.03.2000) US

(71) Applicant: **ABSOLUTEFUTURE, INC.** [US/US]; Suite 1414, 10900 NE 8th Street, Bellevue, WA 98004 (US).

(72) Inventor: **MCNAMARA, Tony**; Suite 1414, 10900 NE 8th Street, Bellevue, WA 98004 (US).

(74) Agents: **PIRIO, Maurice, J.** et al.; Perkins Coie LLP, P.O. Box 1247, Seattle, WA 98111-1247 (US).

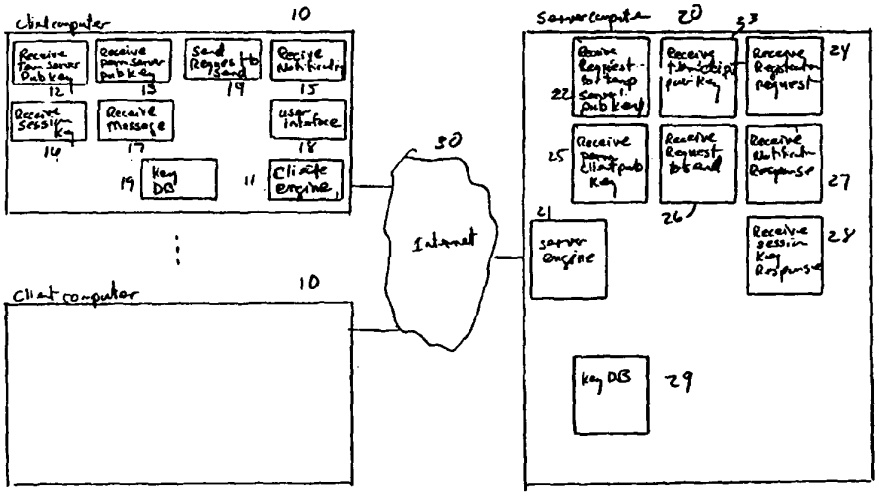
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND SYSTEM FOR COORDINATING SECURE TRANSMISSION OF INFORMATION



(57) Abstract: A method and system for coordinating transmission between a sender and a recipient. In one embodiment, a third party coordinates the distribution of session keys (e.g., symmetric keys) for sender and recipient. A sender computer may generate a session key, encrypt it using a public key of the recipient, and send the encrypted session key to the third party. The third party then forwards the session key to the recipient computer. Upon receiving the session key, the recipient computer decrypts the session key so that the sender and recipient can communicate using a message encrypted with the session key.

WO 01/69843 A2

METHOD AND SYSTEM FOR COORDINATING SECURE TRANSMISSION OF INFORMATION

TECHNICAL FIELD

The described technology generally relates to sending information in a secure manner from one computer system to another.

BACKGROUND

The secure transmission of data is typically accomplished by using a secure channel or by using encryption techniques over a non-secure channel. A secure channel may be established by using a transmission medium that resides totally within a physically secure environment. For example, a government research center may locate a transmission medium (*e.g.*, cabling) between buildings that are fenced off from outside access. An outsider cannot access the information transmitted using that transmission medium, and thus the channel is considered secure. Such secure channels are often restricted to cabling between communication devices that are physically proximate. Encryption techniques allow for the secure transmission of information using a transmission medium that is itself not secure. That is, the encryption techniques secure the information, so the transmission medium does not need to be secure.

Conventional encryption techniques can be categorized as symmetric or asymmetric. Symmetric encryption techniques, such as the Data Encryption Standard ("DES") and the Information Data Encryption Algorithm ("IDEA"), use the same key or password to encrypt and decrypt a message. Before a message can be successfully read by the recipient, the symmetric key needs to be sent from the sender to the recipient. The

symmetric key is typically sent to the recipient separately from the encrypted message. To send the message, the sender encrypts the message using the symmetric key and then transmits the encrypted message to the recipient. The recipient then uses the symmetric key to decrypt the message. A difficulty with symmetric encryption techniques is that symmetric keys are susceptible to being intercepted while enroute to the recipient. Moreover, depending on the technique used to send the symmetric key to the recipient, the sender and recipient may be unaware that the symmetric key has been intercepted.

Asymmetric techniques assign two separate keys, a public key and a private key, to each participant in the secure communications. A message encrypted with a public key can be decrypted with the corresponding private key, and vice versa. A recipient who wants to receive secure messages first generates a public and private key pair. The recipient then publishes its public key for senders to use when sending secure messages to the recipient. To send a message to the recipient, the sender first encrypts the message using the recipient's public key and then sends the encrypted message to the recipient. Upon receipt of the encrypted message, the recipient decrypts the message using its confidential private key. This technique of sending messages is, however, susceptible to identity spoofing. If, however, the sender generates its own public and private key pair and publishes its public key, then the sender could digitally sign the message using its private key. The recipient would decrypt the signature using the public key of the sender. If the decryption is not successful, then the recipient would know that the message was sent by an impostor. A difficulty with asymmetric techniques is that the encrypting and decrypting of messages is computationally expensive. In contrast, symmetric techniques are relatively computationally inexpensive.

To overcome the trade-off between computational expense and security, some systems, such as Pretty Good Privacy ("PGP"), combine

asymmetric and symmetric encryption techniques. Each user of such a system generates a public and private key pair and publishes their public key. When a message is to be sent, such systems generate a symmetric key and encrypt the message using the symmetric key. Such systems then encrypt the symmetric key using the public key of the recipient. When the recipient receives the encrypted message and the encrypted symmetric key, the recipient decrypts the symmetric key using its private key. The recipient then uses the symmetric key to decrypt the message. Because symmetric keys are typically shorter than messages, the overhead of encrypting the symmetric key using the public key is low. Such systems typically reduce the possibility of a "plain-text attack" against the asymmetric encryption of the symmetric key because the symmetric key has no text associated with it. Such systems, however, are susceptible to cryptoanalytical attack techniques associated with symmetric encryption.

It would be desirable to have an encryption technique that would help minimize the difficulties encountered by current encryption techniques.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates the communications between a client computer and a message server computer.

Figure 2 illustrates the communications between a sender computer, a recipient computer, and a message server computer when a message is to be sent from the sender computer to the recipient computer.

Figure 3 is a block diagram illustrating the components of the encryption system in one embodiment.

Figure 4 is a flow diagram illustrating the processing of the receive request for temporary server public key in one embodiment.

Figure 5 is a flow diagram illustrating the processing of the receive temporary client public key component in one embodiment.

Figure 6 is a flow diagram illustrating the processing of the receive permanent client public key component in one embodiment.

Figure 7 is a flow diagram illustrating the processing of the receive registration request component in one embodiment.

Figure 8 is a flow diagram illustrating the processing of the receive request to send component in one embodiment.

Figure 9 is a flow diagram illustrating the processing of the receive notification response component in one embodiment.

Figure 10 is a flow diagram illustrating the processing of the receive session key response component in one embodiment.

Figure 11 is a flow diagram illustrating the processing of the receive temporary server public key component in one embodiment.

Figure 12 is a flow diagram illustrating the processing of the receive permanent server public key component in one embodiment.

Figure 13 is a flow diagram illustrating the processing of the request to send component in one embodiment.

Figure 14 is a flow diagram illustrating the processing of the receive notification component in one embodiment.

Figure 15 is a flow diagram illustrating the processing of the receives session key component in a embodiment.

Figure 16 is a flow diagram illustrating the processing of the receive message component in one embodiment.

DETAILED DESCRIPTION

A method and system for coordinating the transmission of information between a sender and a recipient is provided. In one embodiment, the encryption system provides an asymmetric encryption

mechanism (*e.g.*, a public and private key pair for both the sender and the third party) for communicating between the sender and a third party and an asymmetric encryption mechanism for communicating between the recipient and the third party. When the sender wants to send a message to the recipient, the sender notifies the third party. The third party generates a symmetric key and encrypts the symmetric key using the asymmetric encryption mechanism associated with the sender. The third party then sends that encrypted symmetric key to the sender. (Alternatively, the sender may generate the symmetric key and sent it in encrypted form to the third party so that the recipient, but not the third party, can decrypt the symmetric key.) Upon receiving the encrypted symmetric key, the sender decrypts the symmetric key, encrypts the message with the symmetric key, and sends the encrypted message to the recipient. The third party also encrypts the symmetric key using the asymmetric encryption mechanism associated with the recipient. The third party then sends that encrypted symmetric key to the recipient. Upon receiving the encrypted symmetric key, the recipient decrypts it. When the recipient receives the message from the sender that has been encrypted with the symmetric key, the recipient decrypts the message using the symmetric key. As explained below in more detail, the encryption system takes advantage of the computational efficiency of the symmetric techniques and avoids the problem of having the symmetric key sent with the message (even if encrypted).

In one embodiment, the asymmetric encryption mechanism for communicating between a third party and various clients (*e.g.*, senders and recipients) uses a two-layer asymmetric encryption mechanism. A client who wants to send or receive messages first registers with the third party. During the registration process, both the third party and the client first generate a temporary public and private key pair and exchange their temporary public keys. The third party and the client then generate their own permanent public and private key pair and exchange their permanent public

keys. The third party sends its permanent public key to the client in a form that is encrypted with the temporary public key of the client. When the client receives the permanent public key from the third party, it decrypts the key with its temporary private key. Similarly, the client sends its permanent public key to the third party in a form that is encrypted with the temporary public key of the third party. When the third party receives the permanent public key from the client, it decrypts the key with its temporary private key. The third party can then destroy its permanent public key that it generated for the client, and the client can destroy its permanent public key that it generated for the third party. Eventually, the third party sends symmetric keys to the client, for sending or receiving, by encrypting the symmetric keys with the permanent public key of the client.

Figure 1 illustrates the communications between a client computer and a message server computer during registration. The message server computer functions as the third party to coordinate secure communications between client computers. Initially, the client computer 101 sends a request for a temporary server public key. When the server computer 102 receives the request, it generates a temporary server key pair. The server computer may generate a public and private key pair when it receives the request or may have pre-generated a set of public and private key pairs. The public and private key pairs may be generated by various asymmetric encryption techniques, such as the Rivest, Shamir, and Adelman ("RSA") algorithm. The server computer then sends a temporary server public key to the client computer. When the client computer 103 receives the temporary server public key, it generates a temporary client key pair and sends the temporary client public key to the server computer. The client computer also sends a registration request to the server computer. When the server computer 104 receives the temporary client public key, it stores that key. When the server computer 105 receives the registration request, it generates a permanent server key pair for use in communicating with that client

computer. The server then encrypts the permanent server public key with the temporary client public key for that client computer. The server then sends the encrypted permanent server public key to the client computer. When the client computer 106 receives the encrypted permanent server public key, it decrypts the permanent server public key using its temporary client private key. The client computer then generates a permanent client key pair. The client computer encrypts the permanent client public key with the permanent server public key and sends the encrypted permanent client public key to the server computer. When the server computer 107 receives the permanent client public key, it decrypts the permanent client public key using its permanent server private key. The server computer then stores the permanent client public key in association with the client computer for use in future communications with the client computer. Similarly, the client computer stores the permanent server public key for use in future communications with the server computer. The server computer may generate different sets of temporary and permanent key pairs for each client computer that registers. One skilled in the art will appreciate that the encryption system could register various application programs executing at a client computer. In which case, the registration would be associated with the combination of client computer and application. In addition, the encryption system could associate registrations with users, rather than client computers.

Figure 2 illustrates the communications between a sender computer, a recipient computer, and a message server computer when a message is to be sent from the sender computer to the recipient computer. Assuming that the sender computer and recipient computer are already registered, the sender computer 201 initiates the sending of the message by encrypting using its permanent server public key, a request to send a message to the recipient. The sender computer then sends the encrypted request to the server computer. When the server computer 202 receives the request, it decrypts the request using its permanent server private key for the sender

computer. The request may include the identification of the sender computer in an unencrypted format. Alternatively, the identification may be encrypted with a public key of the server that is known to all sender and recipient computers. The server computer then identifies the recipient from the request and encrypts a notification for the recipient computer using the permanent client public key for the recipient computer, which was received during the registration process for that recipient computer. The server computer then sends the encrypted notification to the recipient computer. When the recipient computer 203 receives the notification, it decrypts the notification using its permanent client private key. The recipient computer then encrypts a response using its permanent server public key and sends the response to the server computer. When the server computer 204 receives response from the recipient computer, it decrypts the response using its permanent server private key for that recipient computer. The server computer then generates a session key (*e.g.*, a symmetric key) for use in encrypting the message that is to be sent from the sender computer and to the recipient computer. The server computer encrypts the session key using the permanent client public key of the recipient computer. The server computer sends the encrypted session key to the recipient computer. When the recipient computer 205 receives the encrypted session key, it decrypts the session key using its permanent client private key. The recipient computer then encrypts a response using its permanent server public key and sends that response to the server computer. When the server computer 206 receives the response from the recipient computer, it encrypts the session key using the permanent client public key of the sender computer. The server computer then sends the encrypted session key to the sender computer. Once the sender computer 207 receives the encrypted session key, it decrypts the session key using its permanent client private key. The sender computer then encrypts a response using its permanent server public key and sends the response to the server computer. When the server computer 208 receives that

response, it decrypts the response using its permanent server private key for the sender computer. To actually send the message, the sender computer 209 encrypts the message with the session key and sends the encrypted message to the recipient computer. When the recipient computer 210 receives the message, it decrypts the message with the session key. The recipient computer then encrypts a response with the session key and sends a response to the sender computer. When the sender computer 211 receives the response from the recipient computer, it decrypts the response using the session key to verify that the recipient computer successfully received the message. One skilled in the art will appreciate that the ordering of the sending of these communications can vary. For example, the symmetric key could be sent to the sender computer before being sent to the recipient computer. Also, the request to send a message could be initiated from a computer other than the sender computer.

In an alternate embodiment, the sender computer, rather than the server computer, generates the session key. In such an embodiment, the server computer sends the permanent client public key of the recipient computer to the sender computer after receiving from the sender computer a request to send a message to the recipient computer. The sender computer creates a session key, encrypts it with the permanent client public key of the recipient computer, and sends the encrypted session key to the server computer. (The sender computer may also encrypt the encrypted session key using the permanent server public key.) Upon receipt of the encrypted session key, the server computer sends the encrypted session key to the recipient computer. Upon receipt of the encrypted session key, the recipient computer decrypts the session key and notifies the server computer, which in turn notifies the sender computer. The sender computer then sends the message to the recipient computer as outlined in 209-211 above. An advantage of this alternate embodiment is that the server computer does not have access to the session key in an unencrypted form. Thus, if the server

computer somehow received the encrypted message, it could not decrypt the message.

Figure 3 is a block diagram illustrating the components of the encryption system in one embodiment. The client computers 310 and the message server computer 320 are interconnected via the Internet 330. The computers may include a central processing unit, memory, input devices (*e.g.*, keyboard and pointing devices), output devices (*e.g.*, display devices), and storage devices (*e.g.*, disk drives). The memory and storage devices are computer-readable media that may contain instructions that implement the encryption system. In addition, the data structures and message structures, including requests and responses, may be stored or transmitted via a data transmission medium such as a signal on a communications link. Various communication channels other than the Internet may be used, such as local area networks, wide area networks, or point-to-point dial-up connections. The client computers include a client engine 311, a receive temporary server public key component 312, a receive permanent server public key component 313, and a send request to send component 314, a receive notification component 315, a receive session key component 316, a receive message component 317, a user interface component 318, and a key database 319. The client engine exchanges communications via the Internet with the message server computer and other client computers. When a communication is received, the client engine invokes the appropriate component for processing the communications. The user interface component provides a mechanism for receiving messages from a sender computer and displaying messages to a user. The mechanism may be similar to a conventional electronic mail application that is adapted to invoke the various components necessary to ensure a secure transmission. The key database contains the permanent client private key of the client computer and the permanent server public key of the message server computer for this client computer. The other components are described in detail below with

reference to the flow diagrams that describe their processing. The server computer includes a server engine 321, a receive request for temporary server key component 322, a receive temporary client public key component 323, a receive registration request component 324, a receive permanent client public key component 325, a receive request to send component 326, a receive notification response component 327, a receive session key response component 328 and a key database 329. The server engine exchanges communications with the client computers and, when a communication is received, invokes the appropriate component for processing. The key database contains a permanent server private key and a permanent client public key for each client computer that has registered with the message server computer. The components of the message server computer are described below in detail with reference to flow diagrams describing their processing.

Figures 4-10 are flow diagrams illustrating processing of the components of the message server computer in one embodiment. Figures 4-7 are flow diagrams illustrating the components of the message server computer used to register a client computer. Figure 4 is a flow diagram illustrating the processing of the receive request for temporary server public key in one embodiment. This component receives a request for a temporary server public key from a client computer. The receipt of the request may be considered to start the registration process. In block 401, the component receives a request along with the identifier of the client computer. The identifier may either be unencrypted or encrypted, for example, with the non-client, computer specific, public key of the server computer. In block 402, the component generates a temporary server key pair for that client computer. In block 403, the component sends the temporary server public key to the client computer. In block 404, the component stores the temporary server private key in association with the identifier of that client computer. The component then completes.

Figure 5 is a flow diagram illustrating the processing of the receive temporary client public key component in one embodiment. This component is invoked when the message server computer receives a client temporary public key. In block 501, the component receives the temporary client public key from the request along with the identifier of the client computer. In block 502, the component stores the temporary client public key in association with the identifier of that client computer.

Figure 6 is a flow diagram illustrating the processing of the receive permanent client public key component in one embodiment. This component is invoked when the message server computer receives a permanent client public key. In block 601, the component receives the permanent client public key and identifier of the client computer. In block 602, the component decrypts the permanent client public key using the permanent server private key for the identified client computer. Alternatively, the permanent client public key may be encrypted with the temporary server public key for that client computer. In this case, the component decrypts the permanent client public key using the temporary server private key for that client computer. In block 603, the component persistently stores the permanent client public key in association with the identified client computer in the key database and then completes.

Figure 7 is a flow diagram illustrating the processing of the receive registration request component in one embodiment. This component is invoked when the message server computer receives a request from a client computer to register after temporary public keys have been exchanged between the client computer and the server computer. In block 701, the component generates a permanent server key pair for the requesting client computer, which is identified in the request. In block 702, the component encrypts the permanent server public key with the temporary client public key for that client computer. In block 703, the component persistently stores the permanent server private key in the key database, associating it with the

identifier of that client computer. In block 704, the component sends the permanent server public key to the client computer and then completes.

Figures 8-11 are flow diagrams illustrating the processing of the server components used when a message is sent from a sender computer to a recipient computer. Figure 8 is a flow diagram illustrating the processing of the receive request to send component in one embodiment. This component is invoked by the message server computer when it receives a request to send a message from a sender computer to a recipient computer. In block 801, the component receives the request along with the identifier of the sender computer. In block 802, the component decrypts the request with the permanent server private key associated with that sender computer that is stored in the key database. In block 803, the component encrypts a notification with the permanent client public key of the recipient computer. In block 804, the component sends the notification to the recipient computer and then completes.

Figure 9 is a flow diagram illustrating the processing of the receive notification response component in one embodiment. This component is invoked by the message server computer when a response to a notification is received from a recipient computer. In block 901, the component receives a response from a recipient computer which includes the identifier of that recipient computer. In block 902, the component decrypts the response using the permanent server private key for that recipient computer that is stored in the key database. In block 903, the component generates a session key. In block 904, the component encrypts the session key with the permanent client public key of the recipient computer. In block 905, the component sends the encrypted session key to the recipient computer and then completes.

Figure 10 is a flow diagram illustrating the processing of the receive session key response component in one embodiment. This component is invoked by the message server computer when a response to

the sending of a session key is received from a sender computer or a recipient computer. In block 1001, the component receives the session key response along with the identifier of the client computer. In block 1002, the component decrypts the response using the permanent server private key of the client computer. In decision block 1003, if the client computer is the recipient computer, then the component continues at block 1004, else the component completes. In block 1004, the component encrypts the session key with the permanent client public key of the sender computer. In block 1005, the component sends the encrypted session key to the sender computer and then completes.

Figures 11-16 are flow diagrams illustrating the processing of a client computer in one embodiment. Figures 11-12 are flow diagrams illustrating the registration process of a client computer. Figure 11 is a flow diagram illustrating the processing of the receive temporary server public key component in one embodiment. The client computer invokes this component when it receives a temporary server public key from the message server computer. In block 1101, the component receives the temporary server public key from the message server computer. In block 1102, the component generates a temporary client key pair. In block 1103, the component sends the temporary client public key to the message server computer. In one embodiment, the component may encrypt the temporary client public key with the temporary server public key. In block 1104, the component sends a registration request to the server and then completes. The sending of the registration request may be temporarily separated from the sending of the temporary client public key.

Figure 12 is a flow diagram illustrating the processing of the receive permanent server public key component in one embodiment. This component is invoked when the client computer receives a permanent server public key from the message server computer. In block 1201, the component receives the permanent server public key from the message server. In block

1202, the component decrypts the permanent server public key using its temporary client private key. In block 1203, the component persistently stores the permanent server public key in the key database. In block 1204, the component generates a permanent client key pair. The component stores the permanent client private key in the key database. In block 1205, the component encrypts the permanent client public key using the permanent server public key. In block 1206, the component sends the permanent client public key to the message server computer and then completes.

Figures 13-16 are flow diagrams illustrating the processing for sending a message from a client computer in one embodiment. Figure 13 is a flow diagram illustrating the processing of the request to send component in one embodiment. This component is invoked when a client computer wants to send a message to a recipient computer. In block 1301, the component encrypts a request using the permanent server public key of the message server computer stored in the key database. In block 1302, the component sends the request to the message server computer and then completes.

Figure 14 is a flow diagram illustrating the processing of the receive notification component in one embodiment. This component is invoked when the client computer receives a notification that it will receive a message from a sender computer. In block 1401, the component receives the notification. In block 1402, the component decrypts the notification using its permanent client private key stored in the key database. In block 1403, the component encrypts a response using the permanent server public key stored in the key database. In block 1405, the component sends the response to the message server computer and then completes.

Figure 15 is a flow diagram illustrating the processing of the receive session key component in one embodiment. This component is invoked when a client computer, sender computer or recipient computer, receives a session key. In block 1501, the component receives a session key from the message server component. In block 1502, the component decrypts

the session key using the permanent client private key stored in the key database. In block 1503, the component encrypts a response with the permanent server public key stored in the key database. In block 1405, the component sends the response to the message server computer. In decision block 1405, if the client computer is the sender computer, then the component continues at block 1506, else the component completes. In block 1506, the component encrypts the message with the session key. In block 1507, the component sends the message to the recipient computer and then completes.

Figure 16 is a flow diagram illustrating the processing of the receive message component in one embodiment. The receive message component is invoked when a client computer receives a message from a sender computer. In block 1601, the component receives the message along with the identifier of the sender computer. In block 1602, the component decrypts the message with the session key for that sender computer. In block 1603, the component encrypts a response using the session key of that sender computer. Alternatively, the response may be encrypted using a permanent client public key of the sender computer, rather than with the session key. In block 1604, the component sends the response to the sender computer and then completes.

From the foregoing, it will be appreciated that although specific embodiments of the encryption system have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. Although the encryption system is described in reference to computers, any type of device that is capable of performing the described processing can be used. For example, the client devices may be personal digital assistants, cell phones, web appliances, and so on. Also, the techniques of the encryption system may be used in conjunction with the Present Detection System to ensure that the recipient computer and optionally, the recipient, is available to receive the

message. Accordingly, the invention is not limited, except by the appended claims.

CLAIMS

1. A computer-based method for coordinating transmitting information between a sender computer and a recipient computer, the method comprising:

providing a sender public key and a recipient public key;

receiving a request to transmit information between the sender computer and the recipient computer;

encrypting a symmetric key using the sender public key and sending the encrypted symmetric key to the sender computer; and

encrypting the symmetric key using the recipient public key and sending the encrypted symmetric key to the recipient computer

whereby the sender computer can receive and decrypt the symmetric key using a sender private key, the recipient computer can receive and decrypt the symmetric key using a recipient private key, and the sender and recipient computers can transmit information that is encrypted using the symmetric key.

2. The method of claim 1 wherein the request to transmit information between the sender computer and the recipient computer is received from the sender computer.

3. The method of claim 1 wherein the request to transmit information between the sender computer and the recipient computer is encrypted using a public key of a message server computer.

4. The method of claim 1 including before sending the encrypted symmetric key to the recipient computer, verifying that the recipient computer is available to receive the encrypted symmetric key.

5. The method of claim 1 including before sending the encrypted symmetric key to the sender computer, verifying that the recipient computer received the encrypted symmetric key.

6. The method of claim 1 wherein the provided sender public key and the recipient public key are used to encrypt multiple symmetric keys for transmitting information between the sender computer and the recipient computer.

7. The method of claim 1 wherein a new symmetric key is used each time a request is received to transmit information between the sender computer and the recipient computer.

8. The method of claim 1 wherein the providing of the sender public key includes sending a temporary server public key to the sender computer and receiving from the sender computer the sender public key encrypted using the temporary server public key, and decrypting the sender public key using a temporary server private key.

9. The method of claim 1 wherein the providing of the recipient public key includes sending a temporary server public key to the recipient computer, receiving from the recipient computer the recipient public key encrypted using the temporary server public key, and decrypting the recipient public key using a temporary server private key.

10. The method of claim 1 wherein the providing of the sender public key includes sending a temporary server public key to the sender computer and receiving from the sender computer the sender public key encrypted using the temporary server public key, and decrypting the sender public key using a temporary server private key; and wherein the providing of the recipient public key includes sending a temporary server public key to the recipient computer and receiving from the recipient computer the recipient public key encrypted using the temporary server public key, and decrypting the recipient public key using a temporary server private key.

11. The method of claim 10 wherein the temporary server public key sent to the sender computer is different from the temporary server public key sent to the recipient computer.

12. The method of claim 1 including ensuring that a recipient is at the recipient computer before sending the encrypted symmetric key to the sender computer.

13. The method of claim 12 including ensuring that the recipient is authorized to receive the information.

14. The method of claim 1 wherein the providing, receiving, encrypting, and sending are performed under control of a message server computer, and whereby the information transmitted between the sender computer and the recipient computer is not sent to the message server computer.

15. The method of claim 1 wherein the public and private keys are RSA based.

16. The method of claim 1 wherein the symmetric key is DES based, IDEA based, or Triple-DES based.

17. A method for coordinating the transmitting of information between a sender and a recipient, the method comprising:

providing a sender asymmetric encryption mechanism for communications between the sender and a third party, and a recipient asymmetric encryption mechanism for communications between the recipient and the third party; and

under control of the third party,

encrypting a symmetric key using the sender asymmetric encryption mechanism and sending the encrypted symmetric key to the sender; and

encrypting the symmetric key using the recipient asymmetric encryption mechanism and sending the encrypted symmetric key to the recipient

whereby the sender can receive and decrypt the symmetric key using the sender asymmetric encryption mechanism, the recipient can receive and decrypt the symmetric key using a recipient asymmetric encryption mechanism, and the sender and recipient can transmit information that is encrypted using the symmetric key.

18. The method of claim 17 including receiving from the sender a request to transmit information to the recipient.

19. The method of claim 18 wherein the request to transmit information to the recipient is encrypted using an asymmetric encryption mechanism.

20. The method of claim 17 including before sending the encrypted symmetric key to the recipient, verifying that the recipient is available to receive the encrypted asymmetric key.

21. The method of claim 17 wherein the provided sender asymmetric mechanism and the provided recipient asymmetric mechanism are used to encrypt multiple symmetric keys for transmitting information between the sender and the recipient.

22. The method of claim 17 wherein a new symmetric key is used whenever a request is received to transmit information between the sender and the recipient.

23. The method of claim 17 wherein the information transmitted from the sender to the recipient is not sent to the third party.

24. The method of claim 17 wherein the asymmetric encryption mechanisms are RSA or Diffie-Hellman based.

25. The method of claim 17 wherein the symmetric key is DES, IDEA, or Triple-DES based.

26. A method in a computer system for coordinating the transmitting of information from a sender computer to a recipient computer,

the sender computer and the recipient computer being client computers, the method comprising:

registering asymmetric encryption data for a plurality of client computers, the registered client computers including the sender computer and the recipient computer;

encrypting symmetric encryption data using the registered asymmetric data for the sender computer and sending the encrypted symmetric encryption data to the sender computer; and

encrypting the symmetric encryption data using the registered asymmetric encryption data for the recipient computer and sending the encrypted symmetric encryption data to the recipient computer.

27. The method of claim 26 whereby the sender computer decrypts the sent symmetric encryption data using its asymmetric encryption data, encrypts the information using the decrypted symmetric encryption data, and sends the encrypted information to the recipient computer.

28. The method of claim 26 wherein the recipient computer decrypts the sent symmetric encryption data using its asymmetric encryption data, receives the encrypted information, and decrypts the received encrypted information using the decrypted symmetric encryption data.

29. The method of claim 26 wherein the registering includes receiving from the client computer the asymmetric encryption data for that client computer.

30. The method of claim 26 wherein the registering includes receiving asymmetric encryption data that is itself encrypted.

31. The method of claim 26 including before sending the encrypted symmetric encryption data to the recipient computer, ensuring that the recipient computer is available to receive the encrypted symmetric encryption data.

32. The method of claim 26 including before sending the encrypted symmetric encryption data to the sender computer, verifying that the recipient computer is available to receive encrypted information from the sender computer.

33. The method of claim 26 including, when multiple instances of information are to be transmitted between the sender computer and the recipient computer, encrypting different symmetric encryption data for each instance with the same asymmetric encryption data of the sender computer and with the same asymmetric encryption data of the recipient computer.

34. The method of claim 26 including ensuring that a person is at the recipient computer before sending the encrypted symmetric encryption data to the recipient computer.

35. The method of claim 34 including ensuring that the person is authorized to receive the information.

36. The method of claim 26 wherein the information is not sent to the computer system.

37. The method of claim 26 wherein the asymmetric encryption data is RSA or Diffie-Hellman based.

38. The method of claim 26 wherein the symmetric encryption data is DES, IDEA, or Triple-DES based.

39. A method in a sender computer for transmitting information to a recipient computer, the method comprising:

establishing an asymmetric encryption mechanism with a server computer;

sending to the server computer a request to send information to the recipient computer;

receiving from the server computer a symmetric key encrypted using the asymmetric encryption mechanism;

decrypting the symmetric key using the asymmetric encryption mechanism;

encrypting information using the decrypted symmetric key; and

sending to the recipient computer the encrypted information.

40. The method of claim 39 wherein the receiving of the symmetric key from the server computer indicates that the recipient computer is available to receive transmitted information.

41. The method of claim 40 wherein the receiving of the symmetric key from the server computer also indicates that a person is at the recipient computer.

42. The method of claim 41 wherein the receiving of the symmetric key from the server computer also indicates that the person is authorized to receive the information.

43. The method of claim 39 wherein the establishing includes sending a public key of the sender computer to the server computer.

44. The method of claim 39 including, for each request sent to the server computer, receiving from the server computer a different symmetric key encrypted with the same asymmetric encryption mechanism.

45. The method of claim 39 wherein the asymmetric encryption mechanism is RSA or Diffie-Hellman based.

46. The method of claim 39 wherein the symmetric key is DES, IDEA, or Triple-DES based.

47. The method of claim 39 wherein the establishing and the sending of the request are temporally separated.

48. The method of claim 39 wherein the establishing is independent of the sending of any request.

49. A method in a recipient computer for receiving information from a sender computer, the method comprising:

establishing an asymmetric encryption mechanism with a server computer;

receiving from the server computer a symmetric key encrypted using the asymmetric encryption mechanism;

decrypting the symmetric key using the asymmetric encryption mechanism;

receiving from the sender computer information encrypted using the symmetric key; and

decrypting the received information using the decrypted symmetric key.

50. The method of claim 49 wherein the establishing includes sending a public key of the recipient computer to the server computer.

51. The method of claim 49 including receiving a different symmetric key encrypted with the asymmetric encryption mechanism for each request received by the server computer to transmit information to the recipient computer.

52. The method of claim 49 including receiving from the server computer an indication of the sender computer that is to send the information encrypted with the received symmetric key.

53. The method of claim 49 wherein the asymmetric encryption mechanism is RSA or Diffie Hellman based.

54. The method of claim 49 wherein the symmetric key is DES, IDEA, or Triple-DES based.

55. The method of claim 49 wherein the establishing and the receiving of the symmetric key are temporally separated.

56. The method of claim 49 wherein the establishing is independent of the receiving of information from a sender computer.

57. The method of claim 49 wherein the establishing is independent of the receiving of any symmetric key from the server computer.

58. A method in a computer system for coordinating transmitting of information from a sender computer to a recipient computer, the sender computer and recipient computer being client computers, the method comprising:

registering asymmetric encryption data for a plurality of client computers, the registered client computers including the sender computer and the recipient computer;

receiving from the sender computer symmetric encryption data that is encrypted by the sender computer using asymmetric encryption data of the recipient computer; and

sending the received symmetric encryption data to the recipient computer so that the sender computer can then send information encrypted with the symmetric encryption data.

59. The method of claim 58 whereby after sending the symmetric encryption data to the recipient computer, the sender computer encrypts the information using the symmetric encryption data and sends the encrypted information to the recipient computer.

60. The method of claim 59 wherein the recipient computer decrypts the sent symmetric encryption data using its asymmetric encryption data, receives the encrypted information, and decrypts the received encrypted information using the decrypted symmetric encryption data.

61. The method of claim 58 wherein the registering includes receiving from each client computer asymmetric encryption data for that client computer.

62. The method of claim 58 including before sending the encrypted symmetric encryption data to the recipient computer, ensuring that the recipient computer is available to receive the encrypted symmetric encryption data.

63. The method of claim 58 including ensuring that a person is at the recipient computer before sending the encrypted symmetric encryption data to the recipient computer.

64. The method of claim 63 including ensuring that the person is authorized to receive the information.

65. The method of claim 58 wherein the information is not sent to the computer system.

66. The method of claim 58 wherein the asymmetric encryption data is RSA or Diffie-Hellman based.

67. The method of claim 58 wherein the symmetric encryption data is DES, IDEA, or Triple-DES based.

68. A method in a sender computer for transmitting information to a recipient computer, the method comprising:

sending to a server computer a symmetric key that has been encrypted using an asymmetric key of the recipient computer;

receiving from the server computer an indication that the recipient computer has received the symmetric key; and

sending to the recipient computer the information encrypted using the symmetric key.

69. The method of claim 68 including before sending to the server computer the symmetric key, receiving from the server computer the asymmetric key of the recipient computer.

70. The method of claim 69 including before receiving the asymmetric key sending to the server computer a request to transmit information to the recipient computer.

71. The method of claim 68 wherein the symmetric key that is encrypted using an asymmetric key of the recipient computer is further encrypted using an asymmetric key of the server computer.

72. The method of claim 68 including before sending the symmetric key to the server computer, registering with the server computer.

73. The method of claim 68 wherein a different symmetric key is used for each transmission to the recipient computer.

74. The method of claim 68 wherein the asymmetric key is RSA or Diffie-Hellman based.

75. The method of claim 68 wherein the symmetric key is DES, IDEA, or Triple-DES based.

76. A method in a recipient computer for receiving information from a sender computer, the method comprising:

receiving from a server computer a symmetric key encrypted using an asymmetric key of the recipient computer;

decrypting the symmetric key using an asymmetric key of the recipient computer;

notifying the server computer that the symmetric key has been received; and

after notifying the server,

receiving from the sender computer information encrypted using the symmetric key; and

decrypting the received information using the decrypted symmetric key.

77. The method of claim 76 wherein the symmetric key is encrypted by the sender computer.

78. The method of claim 76 including receiving a different symmetric key encrypted with the asymmetric key for each request received by the server computer to transmit information to the recipient computer.

79. The method of claim 76 including receiving from the server computer an indication of the sender computer that is to send the information encrypted with the received symmetric key.

80. The method of claim 76 wherein the asymmetric key is RSA or Diffie-Hellman based.

81. The method of claim 76 wherein the symmetric key is DES, IDEA, or Triple-DES based.

82. A computer-readable medium containing instructions for controlling a computer system to coordinate transmitting of information from a sender computer to a recipient computer, the sender computer and recipient computer being client computers, by a method comprising:

receiving a public key for the recipient computer;

sending the received public key to the sender computer;

receiving from the sender computer a symmetric key encrypted using the public key sent to the sender computer; and

sending the received symmetric key to the recipient computer.

83. The computer-readable medium of claim 82 wherein the sender computer encrypts the information using the symmetric key and sends the encrypted information to the recipient computer.

84. The computer-readable medium of claim 83 wherein the recipient computer decrypts the sent symmetric key using its private key, receives the encrypted information, and decrypts the received encrypted information using the decrypted symmetric key.

85. The computer-readable medium of claim 82 including includes receiving from each client computer a public key for that client computer.

86. The computer-readable medium of claim 82 including before sending the encrypted symmetric key to the recipient computer, ensuring that the recipient computer is available to receive the encrypted symmetric key.

87. The computer-readable medium of claim 82 including ensuring that a person is at the recipient computer before sending the encrypted symmetric key to the recipient computer.

88. The computer-readable medium of claim 87 including ensuring that the person is authorized to receive the information.

89. The computer-readable medium of claim 82 wherein the information is not sent to the computer system.

90. The computer-readable medium of claim 82 wherein the public key is RSA or Diffie-Hellman based.

91. The computer-readable medium of claim 82 wherein the symmetric key is DES, IDEA, or Triple-DES based.

92. A computer-readable medium containing instructions for controlling a sender computer to transmit information to a recipient computer, by a method comprising:

sending to a server computer a symmetric key that has been encrypted using a public key of the recipient computer;

receiving from the server computer an indication that the recipient computer has received the symmetric key; and

sending to the recipient computer the information encrypted using the symmetric key.

93. The computer-readable medium of claim 92 including before sending to the server computer the symmetric key, receiving from the server computer the public key of the recipient computer.

94. The computer-readable medium of claim 93 including before receiving the public key sending to the server computer a request to transmit information to the recipient computer.

95. The computer-readable medium of claim 92 wherein the symmetric key that is encrypted using the public key of the recipient computer is further encrypted using a public key of the server computer.

96. The computer-readable medium of claim 92 including before sending the symmetric key to the server computer, registering with the server computer.

97. The computer-readable medium of claim 92 wherein a different symmetric key is used for each transmission to the recipient computer.

98. The computer-readable medium of claim 92 wherein the public key is RSA or Diffie-Hellman based.

99. The computer-readable medium of claim 92 wherein the symmetric key is DES, IDEA, or Triple-DES based.

100. A computer-readable medium containing instructions for controlling a recipient computer to receive information from a sender computer, by a method comprising:

receiving from a server computer a symmetric key encrypted using an public key of the recipient computer;

decrypting the symmetric key using a private key of the recipient computer;

receiving from the sender computer information encrypted using the symmetric key; and

decrypting the received information using the decrypted symmetric key.

101. The computer-readable medium of claim 100 wherein the symmetric key is encrypted by the sender computer.

102. The computer-readable medium of claim 100 including receiving a different symmetric key encrypted with the public key for each transmission by the server computer to the recipient computer.

103. The computer-readable medium of claim 100 including receiving from the server computer an indication of the sender computer that is to send the information encrypted with the received symmetric key.

104. The computer-readable medium of claim 100 wherein the public key is RSA or Diffie Hellman based.

105. The computer-readable medium of claim 100 wherein the symmetric key is DES, IDEA, or Triple-DES based.

106. A server computer for coordinating transmission of information from a sender computer to a recipient computer, comprising:

means for receiving a public key for the recipient computer;

means for sending the received public key to the sender computer;

means for receiving from the sender computer a symmetric key using the public key sent to the sender computer; and

means for sending the received symmetric key to the recipient computer.

107. The server computer of claim 106 including means for registering client computers.

108. The server computer of claim 107 wherein the means for registering computer uses temporary asymmetric keys to transmit permanent asymmetric keys.

109. A server computer for registering public keys, comprising:

means for generating a temporary server public and private key pair;

means for sending the temporary server public key to a client computer;

means for receiving from the client computer a client public key encrypted with the temporary server public key; and

means for decrypting the received client public key using the temporary server private key.

110. A computer system for coordinating secure transmission of information between client computers, comprising:

a component for establishing a different asymmetric encryption mechanism for communication between the computer system and each client computer; and

a component that receives a request to transmit information between requested client computers;

a component that, for each requested client computer, encrypts a symmetric key with the asymmetric encryption mechanism of that client computer and sends the encrypted symmetric key to that client computer.

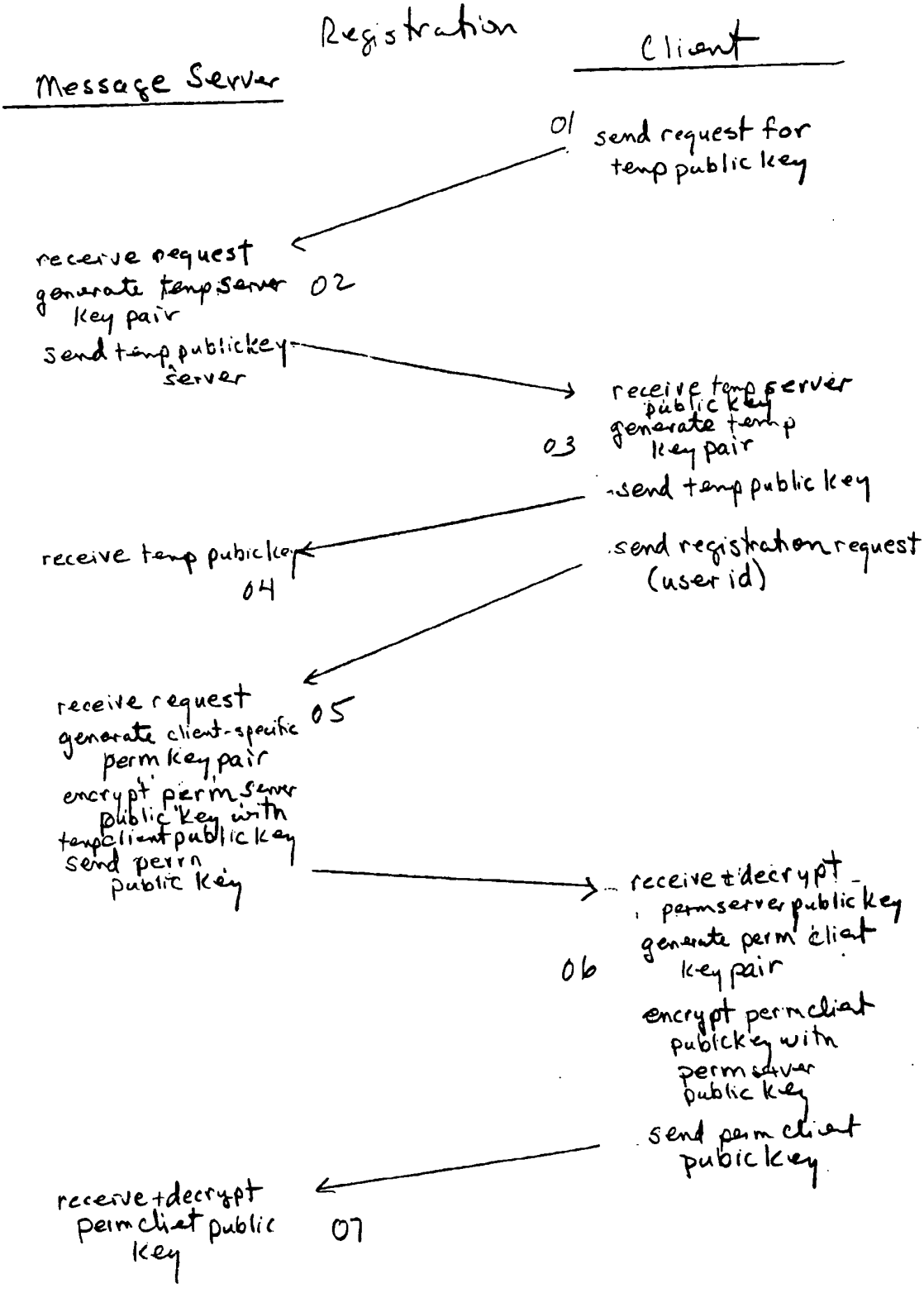

Registration
Message Server
Client
01
send request for temp public key
receive request
generate temp server key pair
02
send temp public key server
receive temp server public key
generate temp key pair
03
send temp public key
receive temp public key
04
send registration request (user id)
receive request
generate client-specific perm key pair
05
encrypt perm server public key with temp client public key
send perm public key
receive + decrypt perm server public key
generate perm client key pair
06
encrypt perm client public key with perm server public key
send perm client public key
receive + decrypt perm client public key
07

Fig 1

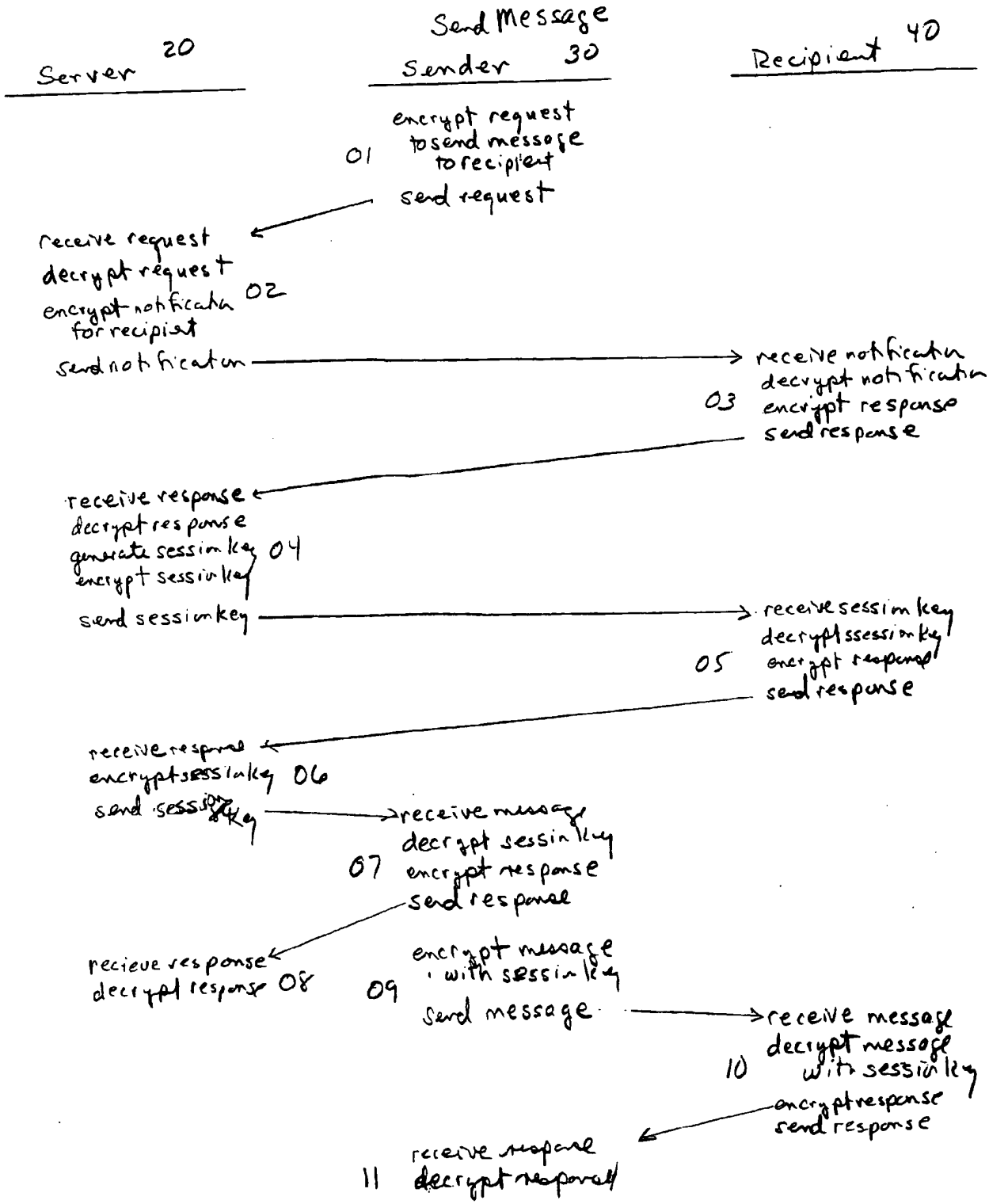

Server 20
Send Message
Sender 30
Recipient 40
encrypt request to send message to recipient
01
send request
receive request
decrypt request
encrypt notification for recipient
02
send notification
receive notification
decrypt notification
03
encrypt response
send response
receive response
decrypt response
generate session key
encrypt session key
04
send session key
receive session key
decrypt session key
05
encrypt response
send response
receive response
encrypt session key
06
send session key
receive message
decrypt session key
07
encrypt response
send response
recieve response
decrypt response
08
09
encrypt message with session key
send message
receive message
decrypt message with session key
10
encrypt response
send response
receive response
11
decrypt response

Fig 2

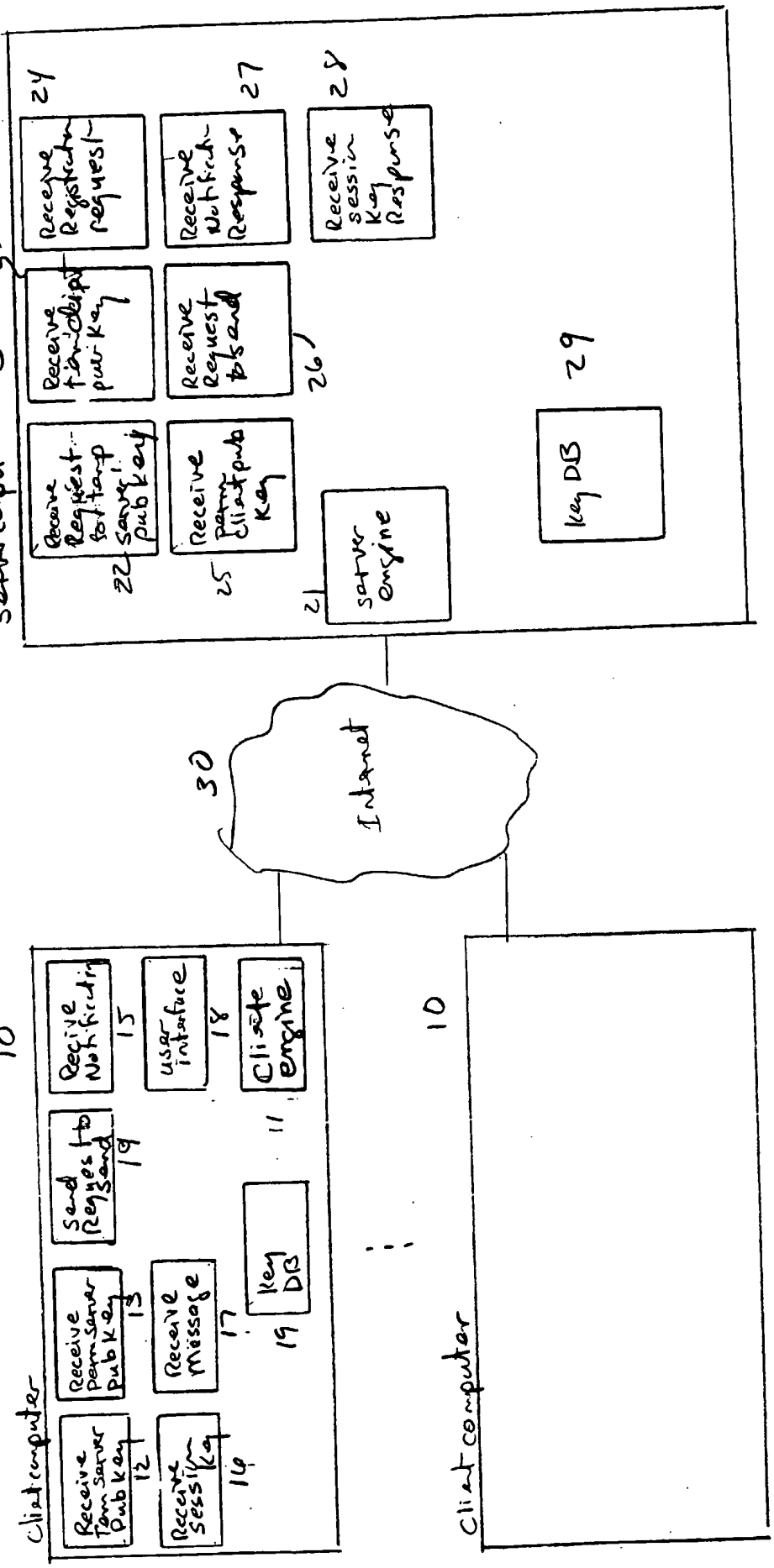

Server computer
20
Receive Registration request
24
Receive Encoded pub key
Receive Request to send
26
Receive Notification Response
27
Receive session key response
28
Receive Request for temp server pub key
22
Receive Perm client pub key
25
server engine
21
key DB
29
Internet
30
Client computer
10
Receive Perm Server Pub Key
12
Receive Session Key
14
Receive Perm Server Pub Key
13
Receive Message
17
Key DB
19
Send Request to Send
19
Receive Notification
15
User interface
18
Client engine
11
Client computer
10

Fig 3

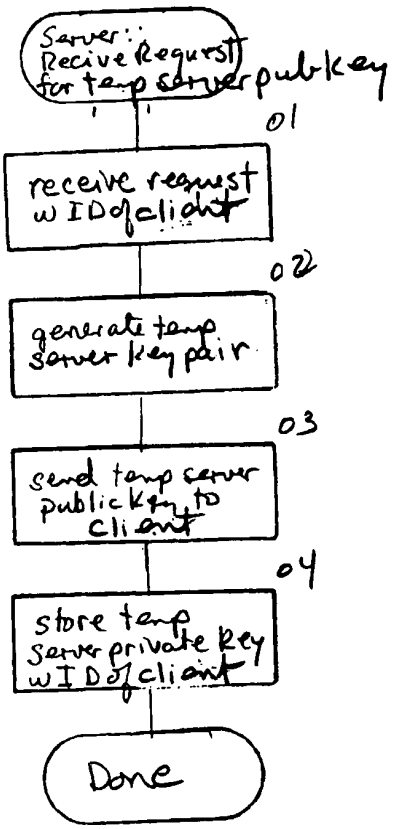

Server:: Recive Request for temp server pub key
01
receive request w ID of client
02
generate temp server key pair
03
send temp server public key to client
04
store temp server private key w ID of client
Done

Fig 4

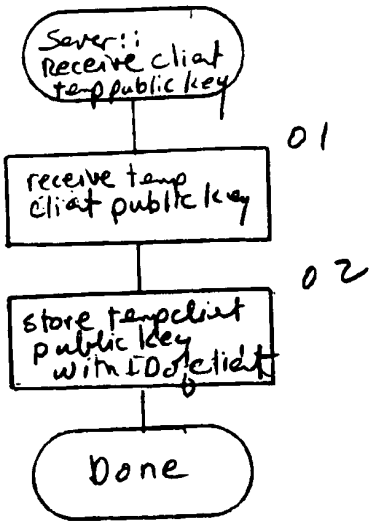

Saver:: Receive cliat temppublic key
01
receive temp cliat public key
02
store tempcliat public key with ID of cliat
Done

Fig 5

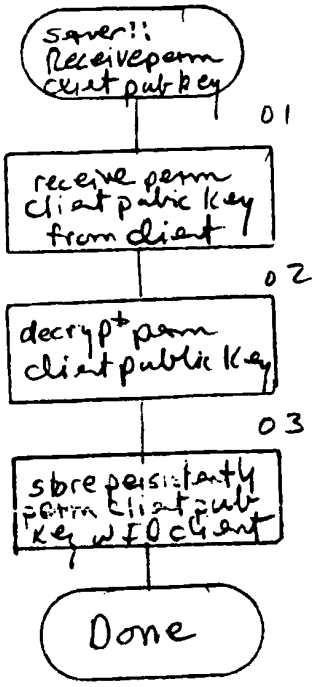

server:
Receive perm
client pub key
01
receive perm
client pblic key
from client
02
decrypt perm
client public key
03
store persistantly
perm client pub
key w/ client
Done

Fig 6

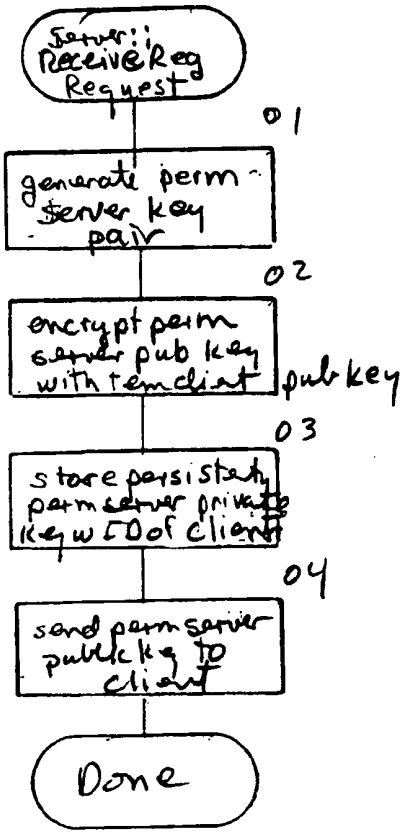

Server: Receive Reg Request
01
generate perm server key pair
02
encrypt perm server pub key with tem client pub key
03
store persistently perm server private key w ID of client
04
send perm server public key to client
Done

Fig 7

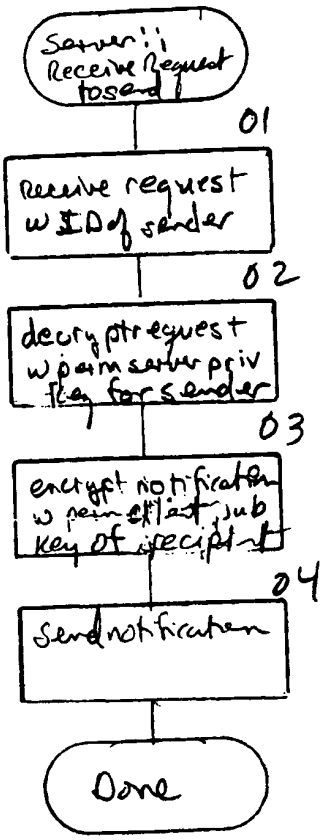

Server: Receive Request to send
01
Receive request w ID of sender
02
decrypt request w perm server priv key for sender
03
encrypt notification w perm client pub key of recipient
04
Send notification
Done

Fig 8

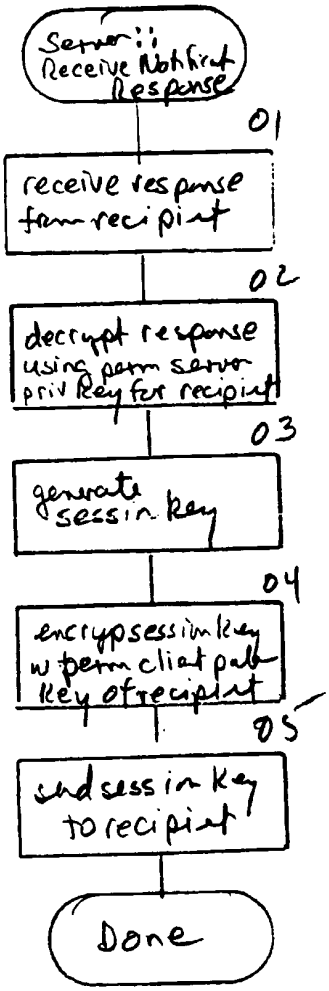

Server:: Receive Notification Response
01
receive response from recipient
02
decrypt response using perm server priv key for recipient
03
generate sessin key
04
encrypt sessin key w perm client public key of recipient
05
send sessin key to recipient
Done

Fig 9

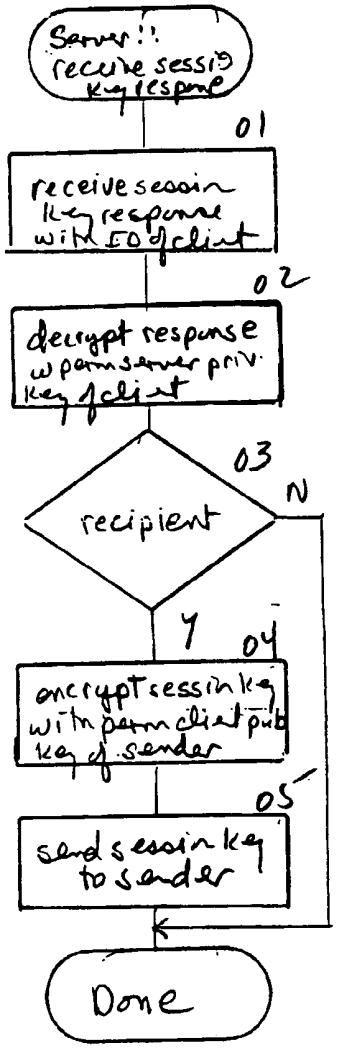

Server!: receive sessio key response
01
receive sessin key response with ID of client
02
decrypt response w perm server priv. key of client
03
recipient
N
Y
04
encrypt sessin key with perm client pub key of sender
05
send sessin key to sender
Done

Fig 10

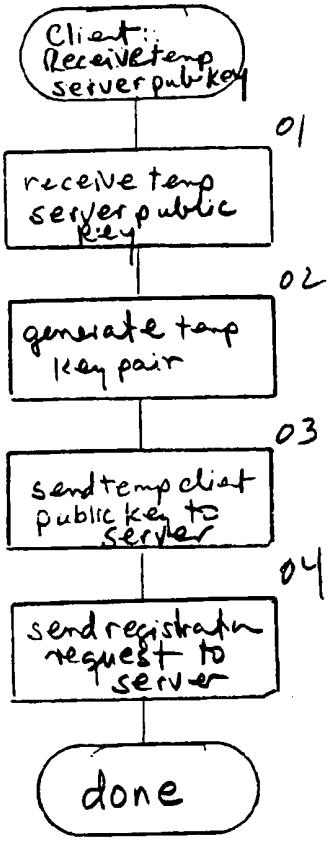

Client: Receive temp server public key
01
receive temp server public key
02
generate temp key pair
03
send temp client public key to server
04
send registration request to server
done

Fig 11

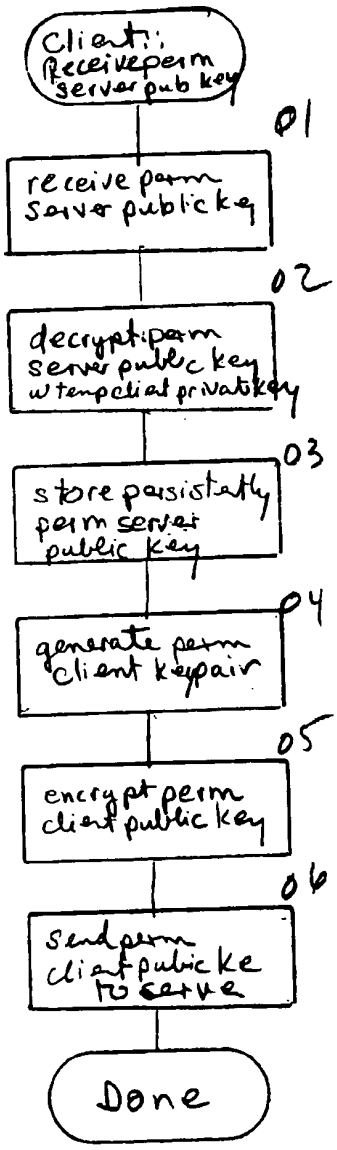

Client: Receive perm server pub key
01
receive perm server public key
02
decrypt perm server public key w temp client private key
03
store persistently perm server public key
04
generate perm client keypair
05
encrypt perm client public key
06
send perm client public ke to serve
Done

Fig 12

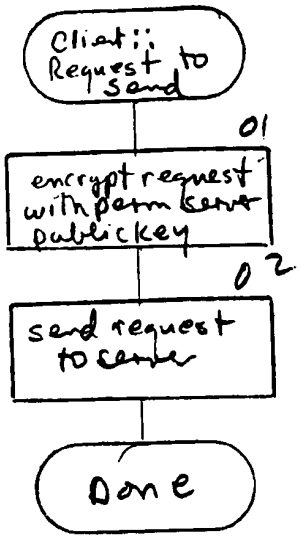

Client Request to send
01
encrypt request with perm serv publickey
02
send request to center
Done

Fig 13

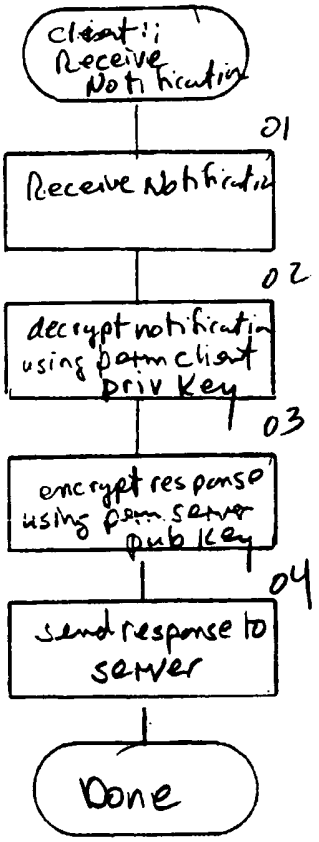

client:
Receive Notification
01
Receive Notification
02
decrypt notification using pgm client priv key
03
encrypt response using pgm server pub key
04
send response to server
Done

Fig 14

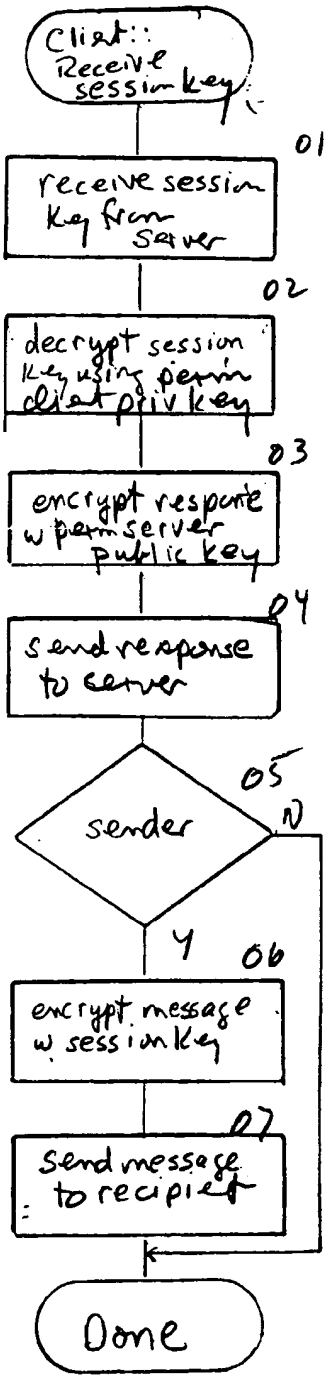

Client:: Receive session key
01
receive session key from server
02
decrypt session key using perm client priv key
03
encrypt response w perm server public key
04
send response to server
05
sender
N
Y
06
encrypt message w session key
07
send message to recipient
Done

Fig 15

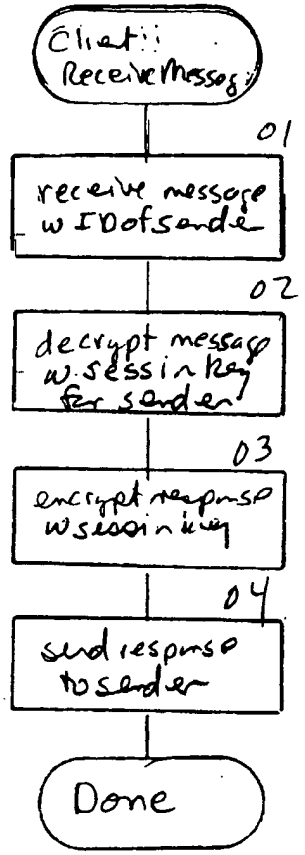

Client: Receive Message
01
receive message w ID of sender
02
decrypt message w. session key for sender
03
encrypt response w session key
04
send response to sender
Done

Fig 16